COMPUTE USER LIBRARY
National Semiconductor
2900 Semiconductor Drive
Santa Clara, CA 95051

NIBASM

A two pass SC/MP assembler written in SC/MP NIBL

BY: ROGER A. MARIN
TEKTRONIX INC.
P.O. BX 500
BEAVERTON,OREGON,U.S.A
97077

29 JULY 1978

CONTENTS

1-5 Program "NIBSAM" discription

6 String compare source listing(a good example).

7 The assembly listing for the compare.

8 The object code dump.

9-11 NIBASM program listing.

REQUIRED HARDWARE:

NIBASM uses NIBL's firmware and so it may be run on either an LCDS or homebrew system configured to run NIBL. NIBASM needs almost 4k of ram or rom memory.This 4k block may be placed on any page 2 thru 7.

As submitted NIBSAM will expect source code text on page 1.That and the fact that it also uses the top 200 bytes of that page make it desireable to have 4k of ram in page 1.To point NIBASM's text cursor to another page change the value of variable "L" on line 110 to equal the address of the first line number byte on the disired page.

SOURCE CODE ENTRY:

Source code is entered and edited using NIBL's text editor. Here's the format:

[line number][a space][label] [opcode] [operand] [comments]

LINE NUMBER- Same rules apply as in regular NIBL programming.
The single space after the line number is'nt optinal (as it is in NIBL).
Labels - Labels may be of any character length.Only the first character defines the label. The first character is also restricted to being alphabetic(A thru Z). A space marks the end of the label.

OPCODE FIELD- This field may contain an opcode,directive,or macro insruction mnemonic.All standard SC/MP mnemonics are supported by NIBASM.

OPERAND FIELD-Pointer designation,auto-indexing mark and displacements may be in any order here. Example:

[disp][ptr][@] OR [ptr][disp][@] OR [@][disp][ptr]

An exeption occurs when a label is in the operand field.The label must be the last item and it must be preceded by a colon[:].

COMMENTS- Comments may appear at any time as long as they are the last item on the line and are preceded by a semi-colon[;].

DISPLACEMENTS AND NUMERIC EXPRESSIONS:

All decimal numbers must be preceeded by a[+] or[-] symbol with no intervening spaces.Also a decimal number must not excede it's equivalent 8-bit binary value.

Example: +0 to +255 or -128 to +127

Hexidecimal values must be preceded by a[#] symbol and have a single byte maximum length. An exeption occurs when applying the [.=] directive;then a 15 bit value is honored.

ASCII CHARACTERS- Ascii characters may be used as displacement or data values. The character must be preceded by a['] symbol

EXPRESSING POINTERS:

Pointers may be expressed by preceding the pointer value with a[P] symbol.

Note that one must not attempt to use apointer value greater than three since NIBASM does'nt check for a maximum limit number on pointers.

Also note that NIBASM will ignore parends around a pointer expression. i.e.:

(P2) is treated the same as P2

DIRECTIVES:

NIBASM supports the following directives: [.END],[.=.],[.=]and[.BYTE].

.END marks the end of the source text

.BYTE (expr) enters a single byte value into the object code.

.= sets the assemblers program counter to the value to the right of [.=].

.=. sets a lable to the left of[.=.] to the assembler's current PC value. If a value is present to the right of [.=.] then the object code counter is incremented by that amount.

THE EQUATE [=] SYMBOL:

Used to the right of a lable to assign an 8-bit expression to the right of the[=] to the lable.

NIBASM MACRO-INSTRUCTIONS

NIBASM contains two macros to save your sanity.They are LDPI(LoaD Pointer IMMEDIATE)and JSR(Jump SubRoutine).

LDPI generates:

```
                LDI(exp high)
                XPAH (ptr)
                LDI(exp low)
                XPAL(ptr)
```

JSR generates:

```
               LDI((exp-1)high)
               XPAH(ptr)
               LDI((exp-1)low)
               XPAL(ptr)
               XPPC(ptr)
```

Where "exp" is a labled address value.

Example:

```
100 STACKPTR   .=. +0
                 -
                 -
                 -
200          LDPI (P2)  :STACKPTR
```

ERROR MESSAGES:

NIBASM flags three error types.They are:

"OP ERROR"-NIBASM found an undefined mnemonic in the opcode field. When this happens NIBASM's program counter and object code address counter are incremented by two.

"JMP DSP ERROR"- this message occurs only when a LABLED pc-relative jump excedes the -128,+127 displacement range.

"UNDFND LBL"- NIBASM looked into it's label table for a value and found it to be set to FFFF(hex);meaning that the label had'nt been previously defind.

NIBASM OBJECT CODE:

When the assembler starts running it will querythe user as to where to place the object code.The value returned will be stored in variable "D" and will serve as a reference for the hex-dump routine as well as the start of the object code block.

After assembly the immediate command "GOTO 1000" will access the dump routine and allow the user a look at the generated object code.

NOTES AND WARNINGS

Although source text must be the first thing present on NIBASM's target page,NIBL text may exist on the same page on the line following the line with the .END directive.

Do not allow source text or object code to over-write the assembler's ram space at the top of page 1. Doing so will have undesired effects.

NIBASM has an addressing limitation in that it will not assemble code exceding the range 0000 to 7FFF.

Lastly -

NIBASM will take some time to do it's job.The assembly listing of the string compare subroutine took approx. 5min on a SC/MP running with a 6mhz external clock.

NIBASM LINE COMMENTS

Let's do the subroutines first.....

#200 to 230 Puts what's in variable "A" into lable table.

#5200 Impliments the "="function(puts the right hand side expression into the lable table).

#6000 to 6030 HEX PRINT SUBR.- Entry value in "X".One or two bytes to print controlled by "Y".

#6100 to 6120 Gets a lable value from table.

#6505 to 6540 HEX ASCII to binary conversion.

#6610 to 6640 DECIMAL ASCII to binary conversion.

#6700 to 6730 Impliments the JSR and LDPI macro's.

Now the main program body.......

#20 Pgrm inputs(note: "W" is the pass counter).

#30 Lable table initialization.

#40 to 70 Loads string compare subroutine into ram.

#110 to 180 Searches for:
    -Source text string
    - Line number in text
    - Lables(defining)
    - The ";" comment flag character.

#4000 to 4510 - Compare "$G" to opcode table (lines 4100 to 4360) and get corresponding hex value.

#4520 to 4532 - Set up case selectors "N" and "J".

#4535 to4700 - Process the key characters in the operand field. The characters looked for are: P # + - : ; '@ and carrage return.

#4710 to 4920 - Manipulate the assemblers PC,store data in object code RAM area and return for a new line.

#1000 to 1020 - This is a program all to its self.It does A hex dump in an endless loop.

```
>LIST
10 ;*********************************
11 ;*                               *
12 ;*   STRING COMPARE FOR NIBASM   *
13 ;*                               *
14 ;*********************************
30 ;
40 ;ON ENTRY FROM NIBL P2=#101C (STRT OF VARIABLES)
60 ;  VAR"Z" MUST BE SET=#00FF
70 ;
100 UPPER =#1F
110 GLOW  = #AB ;DISP FOR LOC. 8100-1
115 INDG  = -3
120 HLOW  = #F8
130 TEMP  = -1
140 CNTR  = -2
150 SLOW  = +36
160 ZLOW  = +50
170 ;
180        .= #1F72
190 ;
200         LDI :UPPER ;SET P1
210         XPAH P1
220         LDI :HLOW
230         XPAL P1
240         LDI #00     ;ZERO LOOP COUNTER.
250         ST(P1):CNTR
255         LDI :GLOW   ;SET UP "$G" PNTR.
256         ST P1:INDG
258 ;
260 NEXT    ILD P2:ZLOW
270         XAE
280         CCL
290         LDI #21
300         CAD (P1)-128;TEST FOR SPACE OR <CR>
310         JP :RETURN   ;TERMINATOR
320         LD P2:SLOW   ;OR VAR"S" NOT ZERO.
330         JNZ:RETURN
340         LD P1-128    ;GET A $H CHAR.
350         ST(P1):TEMP
360         ILD(P1):INDG
370         XAE
380         LD -128(P1)
390         XOR(P1):TEMP
400         ST P2   :SLOW    ;PUT CMPR RESULT IN "S"...
410         JNZ     :RETURN ;END IF NOT ZERO....
420         ILD P1  :CNTR
430         XRI +05          ;OR IF MAX CHAR COUNT
440         JNZ     :NEXT    ;EXCEDED.
450 RET     XPPC P3
460         .END
```

```
>RUN
SCMP ASM
OBJCT CODE RAM=? 7200


 10                    ;**********************************
 11                    ;*                                *
 12                    ;*   STRING COMPARE FOR NIBASM    *
 13                    ;*                                *
 14                    ;**********************************
 30                    ;
 40                    ;ON ENTRY FROM NIBL P2=#101C (STRT OF VARIABLES)
 60                    ; VAR"Z" MUST BE SET=#00FF
 70                    ;
 100    1F              UPPER =#1F
 110    AB              GLOW  = #AB ;DISP FOR LOC. 8100-1
 115    FD              INDG  = -3
 120    F8              HLOW  = #F8
 130    FF              TEMP  = -1
 140    FE              CNTR  = -2
 150    24              SLOW  = +36
 160    32              ZLOW  = +50
 170                    ;
 180                         .= #1F72
 190                    ;
 200 1F72  C4  1F            LDI :UPPER ;SET P1
 210 1F74  35                XPAH P1
 220 1F75  C4  F8            LDI :HLOW
 230 1F77  31                XPAL P1
 240 1F78  C4  00            LDI #00     ;ZERO LOOP COUNTER.
 250 1F7A  C9  FE            ST(P1):CNTR
 255 1F7C  C4  AB            LDI :GLOW   ;SET UP "$G" PNTR.
 256 1F7E  C9  FD            ST P1:INDG
 258                    ;
 260 1F80  AA  32 NEXT       ILD P2:ZLOW
 270 1F82  01                XAE
 280 1F83  02                CCL
 290 1F84  C4  21            LDI #21
 300 1F86  F9  80            CAD (P1)-128;TEST FOR SPACE OR <CR>
 310 1F88  94  19            JP :RETURN  ;TERMINATOR
 320 1F8A  C2  24            LD P2:SLOW  ;OR VAR"S" NOT ZERO.
 330 1F8C  9C  15            JNZ:RETURN
 340 1F8E  C1  80            LD P1-128   ;GET A $H CHAR.
 350 1F90  C9  FF            ST(P1):TEMP
 360 1F92  A9  FD            ILD(P1):INDG
 370 1F94  01                XAE
 380 1F95  C1  80            LD -128(P1)
 390 1F97  E1  FF            XOR(P1):TEMP
 400 1F99  CA  24            ST P2  :SLOW   ;PUT CMPR RESULT IN "S"...
 410 1F9B  9C  06            JNZ    :RETURN ;END IF NOT ZERO....
 420 1F9D  A9  FE            ILD P1 :CNTR
 430 1F9F  E4  05            XRI +05        ;OR IF MAX CHAR COUNT
 440 1FA1  9C  DD            JNZ    :NEXT   ;EXCEDED.
 450 1FA3  3F       RET      XPPC P3
 460           .END

END ASM
```

```
>GOTO 1000
OBJCT CODE

1C20  C4  1F  35  C4  F8  31  C4  00  C9  FE  C4  AB  C9  FD  AA  32
1C30  01  02  C4  21  F9  80  94  19  C2  24  9C  15  C1  80  C9  FF
1C40  A9  FD  01  C1  80  E1  FF  CA  24  9C  06  A9  FE  E4  05  9C
1C50  DD  3F  00
BRK AT 6025
>
```

```
 LIST
20PR"SCMP ASM":PR"OBJCT CODE RAM=";:INPUTD:H=8100:I=7998:W=1
30FORX=0TO51:@(I+X)=#FF:NEXTX:V=8050:Z=-1
40$H="C41F35C4F831C400C9FEC4ABC9FDAA320102C421F9809419C2":GOSUB70:Z=-1
50$H="249C15C180C9FFA9FD01C180E1FFCA249C06A9FEE4059CDD3F":GOSUB70
60H=8184:GOTO110
70FORJ=0TO24:GOSUB6505:@V=B:V=V+1:NEXTJ:RETURN
110 K=D:L=4384:PR""
120X=@L*256+@(L+1):N=L+4:IF(X=-1)OR(W=4)PR"":PR"END ASM":END
140IFW=3PRX;
143IF@N<>59GOTO160
145IFW=3PR"            ";:GOTO150
146IFW=3PR"        ";
150IFW>2PR$(L+3)
152 L=L+@(L+2):IF W=2 W=W+1:GOTO 110
155GOTO120
160IF@N<>32GOSUB200
170N=N+1:IF@N=32GOTO170
175IF@N=59GOTO145
180GOTO4010
200 X=(@N-65)*2:IF(X<0)OR(X>50) GOTO 220
205U=X
210 @(I+X)=@4124:@(I+X+1)=@4125
220 IF @N<>32 N=N+1:GOTO 220
230RETURN
1000PR"OBJCT CODE":PR"":K=D:G=8100:H=8184
1010Y=0:X=K:GOSUB6010:Y=1
1020 FORZ=1TO16:X=@K:GOSUB6010:K=K+1:NEXTZ:PR"":GOTO1010
4010 C=0:G=8100:P=4100:Q=4500:V=0:$G=$N:X=@G
4020IFX<65 P=4330
4030IFX>82P=4270
4040IFX>73 P=4205
4050GOTOP
4100$H="ADD F0":GOTOQ
4105$H="ADE 70":GOTOQ
4110$H="ADI F4":GOTOQ
4115$H="AND D0":GOTOQ
4120$H="ANE 50":GOTOQ
4125$H="ANI D4":GOTOQ
4130$H="CAD F8":GOTOQ
4135$H="CAE 78":GOTOQ
4140$H="CAI FC":GOTOQ
4145$H="CAS 07":GOTOQ
4150$H="CCL 02":GOTOQ
4155$H="CSA 06":GOTOQ
4160$H="DAD E8":GOTOQ
4165$H="DAE 68":GOTOQ
4170$H="DAI EC":GOTOQ
4175$H="DINT 04":GOTOQ
4180$H="DLD B8":GOTOQ
4185$H="DLY 8F":GOTOQ
4190$H="HALT 00":GOTOQ
```

```
 LIST 4195
4195$H="IEN 05":GOTOQ
4200$H="ILD A8":GOTOQ
4205$H="JMP 90":GOTOQ
4210$H="JNZ 9C":GOTOQ
4215$H="JP 94":GOTOQ
4220$H="JZ 98":GOTOQ
4225$H="LDE 40":GOTOQ
4230$H="LDI C4":GOTOQ
4235$H="LDPI FA":GOTOQ
4240$H="LD C0":GOTOQ
4245$H="ORE 58":GOTOQ
4250$H="ORI DC":GOTOQ
4255$H="OR D8":GOTOQ
4260$H="RRL 1F":GOTOQ
4265$H="RR 1E":GOTOQ
4270$H="SCL 03":GOTOQ
4275$H="SIO 19":GOTOQ
4280$H="SRL 1D":GOTOQ
4285$H="SR 1C":GOTOQ
4290$H="ST C8":GOTOQ
4295$H="XAE 01":GOTOQ
4300$H="XOR E0":GOTOQ
4305$H="XPAH 34":GOTOQ
4310$H="XPAL 30":GOTOQ
4315$H="XPPC 3C":GOTOQ
4320$H="XRE 60":GOTOQ
4325$H="XRI E4":GOTOQ
4330$H=".END 10":GOTOQ
4335$H="= FE":GOTOQ
4340$H=".BYTEFF":GOTOQ
4345$H=".=. F2":GOTOQ
4350$H=".= FB":GOTOQ
4355$H="NOP 08":GOTOQ
4360$H="JSR F9":GOTOQ
4365IFW=3 PR"OP ERROR ";
4370GOTO4900
4500S=0:Z=255:LINK8050:IF S<>0 P=P+5:GOTOP
4510GOSUB6505:S=6000:IFB=16W=W+1:GOTO150
4520 V=B:IF(V<32)OR(V>63)AND(V<121) GOTO 4800
4525 G=G+(Z-2):Z=(-1):J=1:IF V>#8F J=2
4527N=2:IFV=255 N=1
4528IFV=254 N=0
4529IFV=242 N=3
4530IFV=251 N=4
4531IFV=#FA N=5:V=0
4532IFV=#F9 N=6:V=0
4535 DO:M=@G:G=G+1:IF(M=13)OR(M=32)GOTO 4700
4540IFM=80 GOSUB6610:V=V+B
4545 IF(M=35)AND(N=4)H=G:GOSUB 6505:A=B*256:GOTO 4547
4546GOTO4560
4547 GOSUB 6505:A=A+B:H=8184:M=13:GOTO 4700
4560IF(M=35)AND(J=1)H=G:GOSUB6505:V=V+B:H=8184:G=G+2
```

```
 LIST4570
4570IFJ=1 GOTO4700
4580IFM=39 C=@G
4590 IF3=(M=64)+(0=(V AND 4))+(V>#BF)V=V+4
4600IF M=35 H=G:GOSUB6505:C=B:H=8184:G=G+2
4610 IF (M=43)OR(M=45) GOSUB 6610:C=B:IF M=45 C=(NOT C)+1
4620 IF M=58 M=@G:GOSUB 6100:C=C+Z:M=13:IF(V>143)AND(V<160) GOTO 4640
4630 GOTO 4700
4640 IF C<=A C=C-(A+2):GOTO 4660
4650 C=C-(A+2)
4660IF((C<-128)OR(C>127))AND(W=3)PR"JMP DSP ERROR";
4700 UNTIL(M=13)OR(M=59):IF(J=2)AND(N=2) GOTO 4900
4710 IF N=1 V=C
4715IFN=3Y=0:X=A:GOSUBS:A=A+C:K=K+C:GOTO146
4720IFN=0GOSUB5200:GOTO146
4730IFN=4GOTO145
4740IF(N=5)OR(N=6)GOSUB6700:GOTO146
4800Y=0:X=A:GOSUBS:A=A+1:X=V:GOSUBS:@K=V:K=K+1
4810 PR"     ";:GOTO 150
4900Y=0:X=A:GOSUBS:A=A+2:X=V:GOSUBS:@K=V:K=K+1
4920X=C:GOSUBS:@K=C:K=K+1:GOTO150
5200X=U:Y=A:A=C AND #7FFF:GOSUB210:A=Y:Y=1:X=C:GOSUBS:RETURN
6000IFW<>3RETURN
6010 $H="    ":$G="0123456789ABCDEF":X=X AND #7FFF
6020@(H+3)=@(G+MOD(X,16)):X=X/16:@(H+2)=@(G+MOD(X,16))
6025IFY=1PR$H;:RETURN
6030X=X/16:@(H+1)=@(G+MOD(X,16)):@H=@(G+X/16):PR$H;:Y=1:RETURN
6100 X=(M-65)*2:@4174=@(I+X):@4175=@(I+X+1)
6110 IF(Z=-1)AND(W=3)PR"UNDFND LBL ";
6120RETURN
6505X=0:B=0
6510 R=@(H+Z+1)-48:IF R>9 R=R-7
6530 B=B*16+R:Z=Z+1:X=X+1:IF X<2 GOTO 6510
6540 RETURN
6610 X=0:B=0:DO:R=@G-48:IF(R<0)OR(R>9) X=1:GOTO 6640
6630 B=B*10+R:G=G+1
6640 UNTIL X=1:RETURN
6700Y=0:X=A:GOSUBS:IFN=6 C=C-1
6705@K=#C4:@(K+1)=@4129:@(K+2)=#34+V
6710@(K+3)=#C4:@(K+4)=@4128:@(K+5)=#30+V:K=K+6:A=A+6
6720IFN=6@K=#3C+V:K=K+1:A=A+1
6730RETURN

>

  PRINT TOP
  12187

 >
```